Sciencia Acta Xaveriana
An International Science Journal
ISSN. 0976-1152

Vol. 8 No. 2
pp. 11-20
September 2017

ALGORITHMS TO FIND VERTEX-TO-CLIQUE MONOPHONIC DISTANCE IN GRAPHS

I. Keerthi Asir[1*], S. Athisayanathan[2] and A. Antonysamy S.J.[3]

[1,2,3]Department of Mathematics, St. Xavier's College (Autonomous),
Palayamkottai - 627 002, Tamil Nadu, India.

[1]E-mail: asirsxc@gmail.com
[2]E-mail: athisxc@gmail.com
[3]E-mail: antonysamy.fr@gmail.com

Received : July 2017 *Accepted : September* 2017

Abstract. In this paper, we find the algorithms for the vertex-to-clique monophonic distance $d_m(i, C)$, the vertex-to-clique monophonic eccentricity $e_{m_1}(i)$, the vertex-to-clique monophonic radius R_{m_1}, the vertex-to-clique monophonic diameter D_{m_1}, the vertex-to-clique monophonic center $C_{m_1}(G)$, and the vertex-to-clique monophonic periphery $P_{m_1}(G)$ of a graph G using BC representation.

Keywords: Vertex-to-clique monophonic distance, vertex-to-clique monophonic eccentricity, vertex-to-clique monophonic center, vertex-to-clique monophonic periphery.

2010 Mathematics Subject Classification: 05C12

1. INTRODUCTION

For basic graph theoretic terminologies, we refer [4, 5]. By a graph $G = (V, E)$, we mean a finite undirected connected simple graph. A clique C of a graph G is a maximal complete subgraph, denoted by its vertices. A chord of a path P is an edge joining two non-adjacent vertices of P. In 1964, Hakimi [6] studied the facility location problems as vertex-to-vertex distance in graphs. For any two vertices u and v in a connected graph G, the distance $d(u, v)$ is the length of a shortest $u-v$ path in G. Also they defined the eccentricity $e(v)$ of a vertex v, the radius r, the diameter d, the center $C(G)$, and the periphery $P(G)$ of a graph G. The distance matrix $d(G) = [d_{ij}]$ of G is a $n \times n$ matrix, where n is the order of G, and $[d_{ij}] = d(v_i, v_j)$, the distance between v_i and v_j in $G (1 \leq i \leq n, 1 \leq j \leq n)$.

In 2005, Chartrand, Escuadro, and Zhang [2] introduced and studied the concepts of detour distance in graphs. For any two vertices u and v in a connected graph G, the detour distance $D(u, v)$ is the length of a longest $u - v$ path in G. Also they defined the detour eccentricity $e_D(v)$ of a vertex v, the detour radius R, the detour diameter D, the detour center $C_D(G)$, and the detour periphery $P_D(G)$ of a graph G. The detour distance matrix $D(G) = [D_{ij}]$ of G is a $n \times n$ matrix, where n is the order of G, and $[D_{ij}] = D(v_i, v_j)$, the detour distance between v_i and v_j in $G(1 \leq i \leq n, 1 \leq j \leq n)$.

Chartrand, Johns and Tian [3]introduced and studied the concepts of an another detour distance in graphs as follows. For any two vertices u and v in a connected graph G, the detour distance $d^*(u, v)$ is the length of a longest induced $u-v$ path in G. That is, a longest $u-v$ path P for which $\langle V(P)\rangle = P$. An induced $u - v$ path of length $d^*(u, v)$ is called a detour path. This detour path contains no chords between any two non-adjacent vertices of P. In 2011, Santhakumaran and Titus [10] rebuilt this detour distance as monophonic distance in graphs. For any two vertices u and v in a connected graph G, the monophonic distance $d_m(u, v)$ is the length of a longest $u - v$ monophonic path in G. Also they defined the monophonic eccentricity $e_m(v)$ of a vertex v, the monophonic radius r_m, the monophonic diameter d_m, the monophonic center $C_m(G)$, and the monophonic periphery $P_m(G)$. The monophonic distance matrix $m(G) = [d_{m_{ij}}]$ of G is a $n\times n$ matrix, where n is the order of G, and $[d_{m_{ij}}] = d_m(v_i, v_j)$, the monophonic distance between v_i and v_j in $G(1 \leq i \leq n, 1 \leq j \leq n)$. For the graph G given in Fig.1.1, the monophonic distance matrix $m(G)$ is

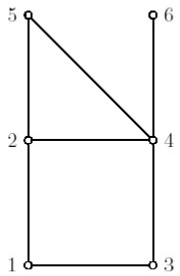

Fig 1.1: G

$$m(G) = \begin{bmatrix} 0 & 1 & 1 & 2 & 3 & 3 \\ 1 & 0 & 2 & 1 & 1 & 2 \\ 1 & 2 & 0 & 1 & 3 & 2 \\ 2 & 1 & 1 & 0 & 1 & 1 \\ 3 & 1 & 3 & 1 & 0 & 2 \\ 3 & 2 & 2 & 1 & 2 & 0 \end{bmatrix}$$

Ashok kumar, Athisayanathan and Antonysamy [1] introduced the algorithms to find vertex-to-clique center in a graph using BC-representation. Correspondingly they defined a method to represent a subset of a set which is called binary count (or BC) representation. That is, $BC(C(i))(1 \leq i \leq n)$ denotes the integer 1 or 0 in the i^{th} place in the BC representation of the clique C in the graph G. For our convenience, we define if $C = (010110)$ then $|C| = 0+1+0+1+1+0 = 3$.

Keerthi Asir and Athisayanathan [7, 9] introduced and studied the concepts of vertex-to-clique detour distance and its algorithms in graphs. Also in [8], Keerthi Asir and Athisayanathan introduced and studied the concepts of vertex-to-clique monophonic distance in graphs. In this paper we introduce and study the algorithms to find the vertex-to-clique monophonic distance $d_m(i, C)$, the vertex-to-clique monophonic eccentricity $e_{m_1}(i)$, the vertex-to-clique monophonic radius R_{m_1}, the vertex-to-clique monophonic diameter D_{m_1}, the vertex-to-clique monophonic center $C_{m_1}(G)$, and the vertex-to-clique monophonic periphery $P_{m_1}(G)$ of a graph G using BC representation. Throughout this paper, G denotes a connected graph with at least two vertices.

2. Vertex-To-Clique Monophonic Distance

First, we introduce an algorithm to find the vertex-to-clique monophonic distance $d_m(i, C)$ between a vertex i and a clique C in a graph G using BC representation.

Definition 2.1. *Let i be a vertex and C a clique in a connected graph G. A vertex-to-clique $i - C$ path P is an $i - j$ path, where j is a vertex in C such that P contains no vertices of C other than j and the $i - C$ path P is said to be an $i - C$ monophonic path if P contains no chords in G. The vertex-to-clique monophonic distance $d_m(i, C)$ is the length of a longest $i - C$ monophonic path in G.*

Algorithm 2.2. *Let G be a non-trivial connected graph with $V = \{1, 2, 3, ..., n\}$ and $\zeta = \{C : C$ is a clique in BC representation$\}$.*

(1) *Let $d_m(G) = [d_{m_{ij}}]$ be the monophonic distance matrix of G.*
(2) *Let $i \in V$*
(3) *Let $C \in \zeta$*
(4) *If $BC(C(i)) = 1$ then $d_m(i, C) = 0$; goto step (10)*
(5) *For $j = 1$ to n*
(6) *If $BC(C(j)) = 0$ then $d_m(i, j) = 0$*
(7) *If $BC(C(j)) = 1$ then $d_m(i, j) = d_{m_{ij}}$*
(8) *Next j*
(9) *Find $d_m(i, C)$*
   - *If $|C| = 2$ and G has more than one $i - C$ monophonic path then $d_m(i, C) = \max\{d_m(i, j) : 1 \le j \le n\}$*
   - *If $|C| > 2$ or G has unique $i - C$ monophonic path then $d_m(i, C) = \max\{d_m(i, j) : 1 \le j \le n\} - 1$*
(10) *Return $d_m(i, C)$*
(11) *Stop*

Theorem 2.3. *For every vertex i and a clique C in a connected graph G, the Algorithm 2.2 finds the vertex-to-clique monophonic distance $d_m(i, C)$.*

Proof. Let G be a graph with $V = \{1, 2, 3, ..., n\}$, $\zeta = \{C : C$ is a clique in BC representation$\}$ and $m(G)$ the monophonic distance matrix of G. Let $i \in V$ and $C \in \zeta$. We consider the following two cases:
Case 1. If $i \in C$ then $BC(C(i)) = 1$ so that the vertex-to-clique monophonic distance $d_m(i, C) = 0$.
Case 2. If $i \notin C$ then $BC(C(i)) = 0$ so that the steps (5) to (8) of the Algorithm 2.2 finds the monophonic distance $d_m(i, j)$ from the vertex i to the vertices $j(1 \leq j \leq n)$ as follows.
Subcase 1 of Case 2. If $j \notin C$ then $BC(C(j)) = 0(1 \leq j \leq n)$ so that the monophonic distance $d_m(i, j) = 0$.
Subcase 2 of Case 2. If $j \in C$ then $BC(C(j)) = 1(1 \leq j \leq n)$ so that the monophonic distance $d_m(i, j) = d_{m_{ij}}$.
Then step (9) of the Algorithm 2.2 finds the vertex-to-clique monophonic distance $d_m(i, C)$ by either $d_m(i, C) = \max\{d_m(i, j) : 1 \leq j \leq n\}$ or $d_m(i, C) = \max\{d_m(i, j) : 1 \leq j \leq n\} - 1$.. □

In the Algorithm 2.2, the step (4) is executed in $O(1)$ time, the steps (5) to (8) are executed in $O(n)$ time, and the step (9) is executed in $O(n)$ time, we have the following theorem.

Theorem 2.4. *The vertex-to-clique monophonic distance $d_m(i, C)$ between the vertex i and the clique C in a graph G can be found in $O(n)$ time.*

Example 2.5. Consider the graph G given in Fig. 1.1, the set ζ of all cliques in G in BC representation is $\zeta = \{(110000), (101000), (010110), (001100), (000101)\}$. Let $D(G)$ be the monophonic distance matrix of G. Now using Algorithm 2.2, let us find the vertex-to-clique monophonic distance $d_m(i, C)$ between the vertex $i = 1$ and the clique $C = \{1, 2\}$. Clearly $BC(C) = (110000)$. Since $BC(C(i)) = 1$, the Algorithm 2.2 returns vertex-to-clique monophonic distance $d_m(i, C) = 0$. Again using the Algorithm 2.2, let us find the vertex-to-clique monophonic distance $d_m(i, C)$ between the vertex $i = 1$ and the clique $C = \{3, 4\}$. Clearly $BC(C) = (001100)$ and $|C| = 2$. Since $BC(C(i)) = 0$, the Algorithm 2.2 finds the vertex-to-clique monophonic distance by $d_m(i, C) = \max\{d_m(i, j) : 1 \leq j \leq n\}$. For the vertices $j = 1, 2, 5, 6$, $BC(C(j)) = 0$ so that $d_m(i, j) = 0$ and also for the vertices $j = 3, 4$, $BC(C(j)) = 1$ so that $d_m(i, 3) = d_{m_{i3}} = 1$ and $d_m(i, 4) = d_{m_{i4}} = 2$. Now the Algorithm 2.2 returns the vertex-to-clique monophonic distance $d_m(i, C) = \max\{d_m(i, j) : 1 \leq j \leq n\} = \max\{d_m(i, 1), d_m(i, 2), d_m(i, 3), d_m(i, 4), d_m(i, 5), d_m(i, 6)\} = \max\{0, 0, 1, 2, 0, 0\} = 2$. Further using the Algorithm 2.2, let us find the vertex-to-clique monophonic distance $d_m(i, C)$ between the vertex $i = 1$ and the clique $C = \{2, 4, 5\}$. Clearly $BC(C) = (010110)$ and $|C| > 2$. Since $BC(C(i)) = 0$, the Algorithm 2.2 finds the vertex-to-clique monophonic distance by $d_m(i, C) = \max\{d_m(i, j) : 1 \leq j \leq n\} - 1$. For the vertices $j = 1, 3, 6$, $BC(C(j)) = 0$ so that $d_m(i, j) = 0$ and also for the vertices $j = 2, 4, 5$, $BC(C(j)) = 1$ so that $d_m(i, 2) = d_{m_{i2}} = 1$, $d_m(i, 4) = d_{m_{i4}} = 2$ and $d_m(i, 5) = d_{m_{i5}} = 3$. Now the Algorithm 2.2 returns the vertex-to-clique monophonic distance $d_m(i, C) = \max\{d_m(i, j) : 1 \leq j \leq n\} - 1 = \max\{d_m(i, 1), d_m(i, 2), d_m(i, 3), d_m(i, 4), d_m(i, 5), d_m(i, 6)\} - 1 = \max\{0, 1, 0, 2, 3, 0\} - 1 = 3 - 1 = 2$.

3. VERTEX-TO-CLIQUE MONOPHONIC ECCENTRICITY

Next, we introduce an algorithm to find the vertex-to-clique monophonic eccentricity $e_{m_1}(i)$ of a vertex i in a graph G using BC representation.

Definition 3.1. *The vertex-to-clique monophonic eccentricity $e_{m_1}(i)$ of a vertex i in a connected graph G is defined as $e_{m_1}(i) = \max\{d_m(i, C) : C \in \zeta\}$, where ζ is the set of all cliques in G.*

Algorithm 3.2. *Let G be a non-trivial connected graph with $V = \{1, 2, 3, ..., n\}$ and $\zeta = \{C : C$ is a clique in BC representation$\}$.*

(1) *Let $\zeta = \{C_1, C_2, ..., C_m\}$.*
(2) *Let $i \in V$*
(3) *For $j = 1$ to m*
(4) *Find $d_m(i, C_j)$, (By Calling Algorithm 2.2)*
(5) *Next j*
(6) *Find $e_{m_1}(i) = \max\{d_m(i, C_j) : 1 \leq j \leq m\}$*
(7) *Return $e_{m_1}(i)$*
(8) *Stop*

Theorem 3.3. *For every vertex i and the set of all cliques ζ in a connected graph G, the Algorithm 3.2 finds the vertex-to-clique monophonic eccentricity $e_{m_1}(i)$.*

Proof. Let G be a non-trivial connected graph with $V = \{1, 2, 3, ..., n\}$ and $\zeta = \{C_1, C_2, \ldots, C_m\}$ the set of all cliques in BC representation in G. Let $i \in V$. Then the step (4) of the Algorithm 3.2 finds the vertex-to-clique monophonic distance $d_m(i, C_j)$ between the vertex i and every clique $C_j (1 \leq j \leq m)$ in G, and the step (6) of the Algorithm 3.2 finds the vertex-to-clique monophonic eccentricity $e_{m_1}(i)$ by $e_{m_1}(i) = \max\{d_m(i, C_j) : 1 \leq j \leq m\}$. □

In the Algorithm 3.2, the step (4) is executed in $O(n)$ time, the steps (3) to (5) are executed in $O(mn)$ time, and the step (6) is executed in $O(m)$ time, we have the following theorem.

Theorem 3.4. *The vertex-to-clique monophonic eccentricity $e_{m_1}(i)$ of a vertex i in a graph G can be found in $O(mn)$ time.*

Example 3.5. For the graph G given in Fig. 1.1, the set ζ of all cliques in G in BC representation is $\zeta = \{(110000), (101000), (010110), (001100), (000101)\}$. Let $i = 1 \in V$. Now using Algorithm 3.2, we find the vertex-to-clique monophonic eccentricity $e_{m_1}(i)$. By calling the algorithm 2.2 m times, the step (4) of Algorithm 3.2 finds the vertex-to-clique monophonic distances $d_m(i, C_1) = 0$, $d_m(i, C_2) = 0$, $d_m(i, C_3) = 2$, $d_m(i, C_4) = 2$, and $d_m(i, C_5) = 2$. Finally the step (6) of Algorithm 3.2 finds the vertex-to-clique monohonic eccentricity $e_{m_1}(i) = \max\{0, 0, 2, 2, 2\} = 2$.

4. VERTEX-TO-CLIQUE MONOPHONIC RADIUS

Next, we introduce an algorithm to find the vertex-to-clique monophonic radius R_{m_1} of a graph G using BC representation.

Definition 4.1. *The vertex-to-clique monophonic radius R_{m_1} of a connected graph G is defined as, $R_{m_1} = rad_{m_1}(G) = \min\{e_{m_1}(i) : i \in V\}$.*

Algorithm 4.2. *Let G be a non-trivial connected graph with $V = \{1, 2, 3, ..., n\}$ and $\zeta = \{C :$ C is a clique in BC representation$\}$.*

(1) *Let $i \in V$*
(2) *For $i = 1$ to n*
(3) *Find $e_{m_1}(i)$, (By Calling Algorithm 3.2)*
(4) *Next i*
(5) *Find $R_{m_1} = \min\{e_{m_1}(i) : 1 \leq i \leq n\}$*
(6) *Return R_{m_1}*
(7) *Stop*

Theorem 4.3. *For a connected graph G, the Algorithm 4.2 finds the vertex-to-clique monophonic radius R_{m_1} of G.*

Proof. Let G be a non-trivial connected graph with $V = \{1, 2, 3, ..., n\}$ and $i \in V$. Then the steps (2) to (4) of the Algorithm 4.2 finds the vertex-to-clique monophonic eccentricity $e_{m_1}(i)$ for every vertex i, and the step (5) of the Algorithm 4.2 finds the vertex-to-clique monophonic radius R_{m_1} of G by $R_{m_1} = \min\{e_{m_1}(i) : 1 \leq i \leq n\}$. □

In the Algorithm 4.2, the step (3) is executed in $O(mn)$ time, the steps (2) to (4) are executed in $O(mn^2)$ time, and the step (5) is executed in $O(n)$ time, we have the following theorem.

Theorem 4.4. *The vertex-to-clique monophonic radius R_{m_1} of G can be found in $O(mn^2)$ time.*

Example 4.5. For the graph G given in Fig. 1.1, the set ζ of all cliques in G in BC representation is $\zeta = \{(110000), (101000), (010110), (001100), (000101)\}$. Let $i = 1 \in V$. Now using Algorithm 4.2, we find the vertex-to-clique monophonic radius R_{m_1} of G. By calling the algorithm 3.2 n times, the step (3) of Algorithm 4.2 finds the vertex-to-clique monophonic eccentricities $e_{m_1}(1) = 2$, $e_{m_1}(2) = 2$, $e_{m_1}(3) = 2$, $e_{m_1}(4) = 2$, $e_{m_1}(5) = 3$ and $e_{m_1}(6) = 3$. Finally step (5) of Algorithm 4.2 finds the vertex-to-clique monophonic radius $R_{m_1} = \min\{2, 2, 2, 2, 3, 3\} = 2$.

5. Vertex-To-Clique Monophonic Center

Next, we introduce an algorithm to find the vertex-to-clique monophonic center $C_{m_1}(G)$ of a graph G using BC representation.

Definition 5.1. *Let G be a connected graph. A vertex i in G is called a vertex-to-clique monophonic central vertex if $e_{m_1}(i) = R_{m_1}$ and the vertex-to-clique monophonic center $C_{m_1}(G)$ of G is defined as, $C_{m_1}(G) = Cen_{m_1}(G) = \langle\{i \in V : e_{m_1}(i) = R_{m_1}\}\rangle$.*

Algorithm 5.2. *Let G be a non-trivial connected graph with $V = \{1, 2, 3, ..., n\}$ and $\zeta = \{C :$ C is a clique in BC representation$\}$.*

(1) *Let $\zeta = \{C_1, C_2, ..., C_m\}$.*

(2) *Let* $C_{m_1}(G) = \langle\phi\rangle$
(3) *For* $i = 1$ *to* n
(4) *Find* $e_{m_1}(i)$*, (By Calling Algorithm 3.2)*
(5) *Next* i
(6) *Find* R_{m_1}*, (By Calling Algorithm 4.2)*
(7) *For* $i = 1$ *to* n
(8) *If* $e_{m_1}(i) = R_{m_1}$ *then* $C_{m_1}(G) = C_{m_1}(G) \cup \{i\}$
(9) *Next* i
(10) *Stop*

Theorem 5.3. *For a connected graph G, the Algorithm 5.2 finds the vertex-to-clique monophonic center* C_{m_1} *of G.*

Proof. Let G be a non-trivial connected graph with $V = \{1, 2, 3, ..., n\}$ and $\zeta = \{C_1, C_2, \ldots, C_m\}$ the set of all cliques in BC representation in G. Then the steps (3) to (5) of the Algorithm 5.2 finds the vertex-to-clique monophonic eccentricity $e_{m_1}(i)$ for every vertex $i \in V(1 \leq i \leq n)$, the step (6) of the Algorithm 5.2 finds the vertex-to-clique monophonic radius R_{m_1} of G by $R_{m_1} = \min\{e_{m_1}(i) : 1 \leq i \leq n\}$, and the steps (7) to (9) of the Algorithm 5.2 finds the vertex-to-clique monophonic center $C_{m_1}(G)$ of G by $C_{m_1}(G) = Cen_{m_1}(G) = \langle\{i \in V : e_{m_1}(i) = R_{m_1}\}\rangle$. □

In the Algorithm 5.2, the step (4) is executed in $O(mn)$ time, the steps (3) to (5) are executed in $O(mn^2)$ time, the step (6) is executed in $O(n)$ time, and the steps (7) to (9) are executed in $O(n)$ time, we have the following theorem.

Theorem 5.4. *The vertex-to-clique monophonic center* $C_{m_1}(G)$ *of G can be found in* $O(mn^2)$ *time.*

Example 5.5. For the graph G given in Fig. 1.1, the set ζ of all cliques in G in BC representation is $\zeta = \{(110000), (101000), (010110), (001100), (000101)\}$. Let $i = 1 \in V$. Now using Algorithm 5.2, we find the vertex-to-clique monophonic center $C_{m_1}(G)$. By calling the algorithm 3.2 n times, the step (4) of Algorithm 5.2 finds the vertex-to-clique monophonic eccentricities $e_{m_1}(1) = 2$, $e_{m_1}(2) = 2$, $e_{D1}(3) = 2$, $e_{m_1}(4) = 2$, $e_{m_1}(5) = 3$ and $e_{m_1}(6) = 3$. By calling the algorithm 4.2 n times, step (6) of Algorithm 5.2 finds the vertex-to-clique monophonic radius $R_{m_1} = \min\{2, 2, 2, 2, 3, 3\} = 2$. Finally step (8) of Algorithm 5.2 finds the vertex-to-clique monophonic center $C_{m_1}(G) = \langle\{i \in V : e_{m_1}(i) = R_{m_1}\}\rangle = \langle\{1, 2, 3, 4\}\rangle$.

6. Vertex-To-Clique Monophonic Diameter

Next, we introduce an algorithm to find the vertex-to-clique monophonic diameter D_{m_1} of a graph G using BC representation.

Definition 6.1. *The vertex-to-clique monophonic diameter* D_{m_1} *of a connected graph G is defined as,* $D_{m_1} = diam_{m_1}(G) = \max\{e_{m_1}(i) : i \in V\}$.

Algorithm 6.2. *Let G be a non-trivial connected graph with* $V = \{1, 2, 3, ..., n\}$ *and* $\zeta = \{C : C \text{ is a clique in BC representation}\}$.

(1) *Let* $i \in V$
(2) *For* $i = 1$ *to* n
(3) *Find* $e_{m_1}(i)$*, (By Calling Algorithm 3.2)*
(4) *Next* i
(5) *Find* $D_{m_1} = \max\{e_{m_1}(i) : 1 \leq i \leq n\}$
(6) *Return* D_{m_1}
(7) *Stop*

Theorem 6.3. *For a connected graph G, the Algorithm 6.2 finds the vertex-to-clique monophonic diameter* D_{m_1} *of G.*

Proof. Let G be a non-trivial connected graph with $V = \{1, 2, 3, ..., n\}$ and $i \in V$. Then the steps (2) to (4) of the Algorithm 6.2 finds the vertex-to-clique monophonic eccentricity $e_{m_1}(i)$ for every vertex i, and the step (5) of the Algorithm 6.2 finds the vertex-to-clique monophonic diameter D_{m_1} of G by $D_{m_1} = \max\{e_{m_1}(i) : 1 \leq i \leq n\}$. □

In the Algorithm 6.2, the step (3) is executed in $O(mn)$ time, the steps (2) to (4) are executed in $O(mn^2)$ time, and the step (5) is executed in $O(n)$ time, we have the following theorem.

Theorem 6.4. *The vertex-to-clique monophonic diameter* D_{m_1} *of G can be found in* $O(mn^2)$ *time.*

Example 6.5. For the graph G given in Fig. 1.1, the set ζ of all cliques in G in BC representation is $\zeta = \{(110000), (101000), (010110), (001100), (000101)\}$. Let $i = 1 \in V$. Now using Algorithm 6.2, we find the vertex-to-clique monophonic diameter D_{m_1} of G. By calling the algorithm 3.2 n times, the step (3) of Algorithm 6.2 finds the vertex-to-clique monophonic eccentricities $e_{m_1}(1) = 2$, $e_{m_1}(2) = 2$, $e_{m_1}(3) = 2$, $e_{m_1}(4) = 2$, $e_{m_1}(5) = 3$ and $e_{m_1}(6) = 3$. Finally step (5) of Algorithm 6.2 finds the vertex-to-clique monophonic diameter $D_{m_1} = \max\{2, 2, 2, 2, 3, 3\} = 3$.

7. Vertex-To-Clique Monophonic Periphery

Next, we introduce an algorithm to find the vertex-to-clique monophonic periphery $P_{m_1}(G)$ of a graph G using BC representation.

Definition 7.1. *Let G be a connected graph. A vertex* i *in G is called a vertex-to-clique monophonic peripheral vertex if* $e_{m_1}(i) = D_{m_1}$ *and the vertex-to-clique monophonic periphery* $P_{m_1}(G)$ *of G is defined as,* $P_{m_1}(G) = Per_{m_1}(G) = \langle\{i \in V : e_{m_1}(i) = D_{m_1}\}\rangle$.

Algorithm 7.2. *Let G be a non-trivial connected graph with* $V = \{1, 2, 3, ..., n\}$ *and* $\zeta = \{C :$ C *is a clique in BC representation*$\}$.

(1) *Let* $\zeta = \{C_1, C_2, ..., C_m\}$.
(2) *Let* $P_{m_1}(G) = \langle\phi\rangle$
(3) *For* $i = 1$ *to* n
(4) *Find* $e_{m_1}(i)$*, (By Calling Algorithm 3.2)*
(5) *Next* i

(6) *Find* D_{m_1}, *(By Calling Algorithm 6.2)*
(7) *For* $i = 1$ *to* n
(8) *If* $e_{m_1}(i) = D_{m_1}$ *then* $P_{m_1}(G) = P_{m_1}(G) \cup \{i\}$
(9) *Next* i
(10) *Stop*

Theorem 7.3. *For a connected graph* G*, the Algorithm 7.2 finds the vertex-to-clique monophonic periphery* P_{m_1} *of* G*.*

Proof. Let G be a non-trivial connected graph with $V = \{1, 2, 3, ..., n\}$ and $\zeta = \{C_1, C_2, \ldots, C_m\}$ the set of all cliques in BC representation in G. Then the steps (3) to (5) of the Algorithm 7.2 finds the vertex-to-clique monophonic eccentricity $e_{m_1}(i)$ for every vertex $i \in V(1 \leq i \leq n)$, the step (6) of the Algorithm 7.2 finds the vertex-to-clique monophonic diameter D_{m_1} of G by $D_{m_1} = \max\{e_{m_1}(i) : 1 \leq i \leq n\}$, and the steps (7) to (9) of the Algorithm 7.2 finds the vertex-to-clique monophonic periphery $P_{m_1}(G)$ of G by $P_{m_1}(G) = Per_{m_1}(G) = \langle\{i \in V : e_{m_1}(i) = D_{m_1}\}\rangle$. □

In the Algorithm 7.2, the step (4) is executed in $O(mn)$ time, the steps (3) to (5) are executed in $O(mn^2)$ time, the step (6) is executed in $O(n)$ time, and the steps (7) to (9) are executed in $O(n)$ time, we have the following theorem.

Theorem 7.4. *The vertex-to-clique monophonic periphery* $P_{m_1}(G)$ *of* G *can be found in* $O(mn^2)$ *time.*

Example 7.5. For the graph G given in Fig. 1.1, the set ζ of all cliques in G in BC representation is $\zeta = \{(110000), (101000), (010110), (001100), (000101)\}$. Let $i = 1 \in V$. Now using Algorithm 7.2, we find the vertex-to-clique monophonic periphery $P_{m_1}(G)$. By calling the algorithm 3.2 n times, the step (4) of Algorithm 7.2 finds the vertex-to-clique monophonic eccentricities $e_{m_1}(1) = 2$, $e_{m_1}(2) = 2$, $e_{m_1}(3) = 2$, $e_{m_1}(4) = 2$, $e_{m_1}(5) = 3$ and $e_{m_1}(6) = 3$. By calling the algorithm 6.2 n times, step (6) of Algorithm 7.2 finds the vertex-to-clique monophonic diameter $D_{m_1} = \max\{2, 2, 2, 2, 3, 3\} = 3$. Finally step (8) of Algorithm 7.2 finds the vertex-to-clique monophonic periphery $P_{m_1}(G) = \langle\{i \in V : e_{m_1}(i) = D_{m_1}\}\rangle = \langle\{5, 6\}\rangle$.

8. Open Problem

Problem 8.1. *Discuss the algorithm for the vertex-to-clique monophonic center of every connected graph* G *lies in a single block of* G*.*

Problem 8.2. *Discuss the algorithm for the vertex-to-clique monophonic self-centered graph.*

9. Conclusion

In a social network a clique represents a group of individuals having a common interest. Thus the centrality with respect to cliques have interesting applications in social networks. For example if one wants to design a security based communication network, the monophonic concepts play a vital role. These concepts have interesting applications in channel assignment problems in radio technologies and capture different aspects of certain molecular problems in theoretical chemistry. In the case of designing the channel for a communication network, although maximum number of vertices are covered by the network when considering monophonic paths, some of the edges (chords) may be left out. This drawback is rectified in the case of monophonic distance so that considering vertex-to-clique monophonic distance is more advantageous to real life application of communication networks.

ACKNOWLEDGMENTS

The author wish to thank God.

REFERENCES

[1] A. Ashok Kumar, S. Athisayanathan and A. Antonysamy, *Algorithms to find Vertex-to-Clique Center in a Graph using BC-representation*, Int. J. Advanced Networking and Applications, **4**, (2013), 1809-1811.

[2] G. Chartrand, H. Escuadro and P. Zhang, *Detour Distance in Graphs*, J. Combin. Math. Combin. Comput., **53**, (2005), 75-94.

[3] G. Chartrand, G. L. Johns and S. Tian, *Detour Distance in Graphs*, Annals of Discrete Mathematics, **55**, (1993), 127-136.

[4] G. Chartrand, Ortrud R. Oellermann, *Applied and Algorithmic Graph Theory*, McGrraw-Hill International Editions (1993).

[5] G. Chartrand and P. Zhang, *Introduction to Graph Theory*, Tata McGraw-Hill, New Delhi, 2006.

[6] S. L. Hakimi, *Optimum location of switching centers and absolute centers and medians of a graph*, Operations Research, **12**, (1964).

[7] I. Keerthi Asir and S. Athisayanathan, *Vertex-to-Clique Detour Distance in Graphs*, Journal of Prime Research In Mathematics, **12**, (2016), 45-59.

[8] I. Keerthi Asir and S. Athisayanathan, *Vertex-to-Clique Monophonic Distance in Graphs*, Ars Combinatoria, (Accepted).

[9] I. Keerthi Asir, S. Athisayanathan and A. Antonysamy, *Algorithms to Find Vertex-to-Clique Detour Distance in Graphs*, TWMS J. of Apl. Eng. Math., (Accepted).

[10] A. P. Santhakumaran and P. Titus, *Monophonic Distance in Graphs*, Discrete Math. Algorithms Appl., **3**, (2011), 159-169.